2

COMPUTER SCIENCE
TECHNICAL REPORT SERIES

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742

TR-1179
DAAG-53-76C-0138

June, 1982

CONTOUR-BASED MOTION ESTIMATION

Larry S. Davis
Zhongquan Wu
Hanfang Sun

Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

ABSTRACT

This paper introduces a contour-based approach to motion estimation. It is based on first computing motion at image corners, and then propagating the corner motion estimates along the principal contours in the image based on a local 2½D motion assumption. The resu.ts of several experiments are presented.

The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (DARPA Order 3206) is gratefully acknowledged, as is the help of Janet Salzman in preparing this paper. This paper is based in part on two earlier reports (TR's 1130 and 1132).

1. Introduction

The earliest problem that arises in the analysis of time-varying images is the detection of moving image elements (edge, regions) and the computation of the image velocity (optic flow) of those elements. A variety of computational schemes have been proposed to solve this problem. In a recent survey, Ullman [1] broadly classifies these as intensity-based and token-matching schemes.

An important class of intensity-based schemes takes advantage of the relationship between the temporal and spatial gradient of any continuous and differentiable image property which is invariant to small changes in perspective. For example, if we assume that the intensity, I, satisfies these properties, the relationship

$$-I_t = uI_x + vI_y \qquad (1)$$

can be used to determine velocity. Here, I_t is the temporal intensity gradient, I_x and I_y the x and y components of the spatial intensity gradient, and u and v the x and y components of image velocity. Measuring I_t, I_x, I_y from an image sequence establishes a linear constraint on the x and y velocity components. A single velocity estimate can be computed by spatially combining the constraints using e.g., Hough transforms [2], least-squares methods [3] or minimization techniques [4]. All of these techniques suffer from certain disadvantages. The Hough-transform and minimization techniques assume that image velocity is uniform over large parts of the image, and the least-squares method further assumes that the constraint equations determined for nearby points are independent - an assumption that is violated by the spatial integration required to compute spatial derivatives.

In this paper we develop a contour-based approach to motion estimation at a small set of image points at which it is possible, in principle, to unambiguously determine image velocity. Specifically, corners have the property that their motion can be directly computed based only on measurements made at the corner (in practice, of course, one must examine a small neighborhood of the corner). Another important property of corners is that they can be safely regardcd as projections of scene features whose general appearance is invariant to rigid motion - e.g., an image corner may be the projection of the vertex of a polyhedron, or of a curvature discontinuity on the boundary of a surface marking. The second step involves propagating these velocity estimates to a larger number of picture points. This is based on the assumption that the image motion is locally a rigid two-dimensional motion. Given this assumption, the velocity at one point on a contour and the normal component at a neighboring point can be combined to compute the actual velocity at the neighboring point.

2. Estimating motion at corners

The motion of a corner can be computed in a variety of ways. Section 2.1 describes an approach based on temporal intensity changes along lines parallel to the sides of the corner. Section 2.2 discusses a second technique which combines normal vectors in a small neighborhood of the corner along the contour. It is similar to the velocity estimation algorithm in Horn and Schunck [4]. We should also point out that Nagel [5] has recently proposed a corner velocity estimation algorithm based on a differential approach (i.e., a Taylor series expansion of the image function in the neighborhood of the corner truncated after the second-order terms.).

2.1 A structural approach

This subsection presents a structural approach to corner motion estimation. We first describe velocity computation for the case of translation motion, and then consider translation combined with rotation.

Suppose that a corner simply translates from point C_0 to C_1 between two frames t_0 and t_1 (see Figure 1). Let O' be a point on the bisector of PC_0R and let O'A and O'B be lines parallel to C_0P and C_0R, respectively at some unit distance from C_0P and C_0R. Suppose that $|O'A| = |O'B| = 1+m$, for some constant m. Finally, assume that the intensity inside the corner is 1 and outside the corner is 0.

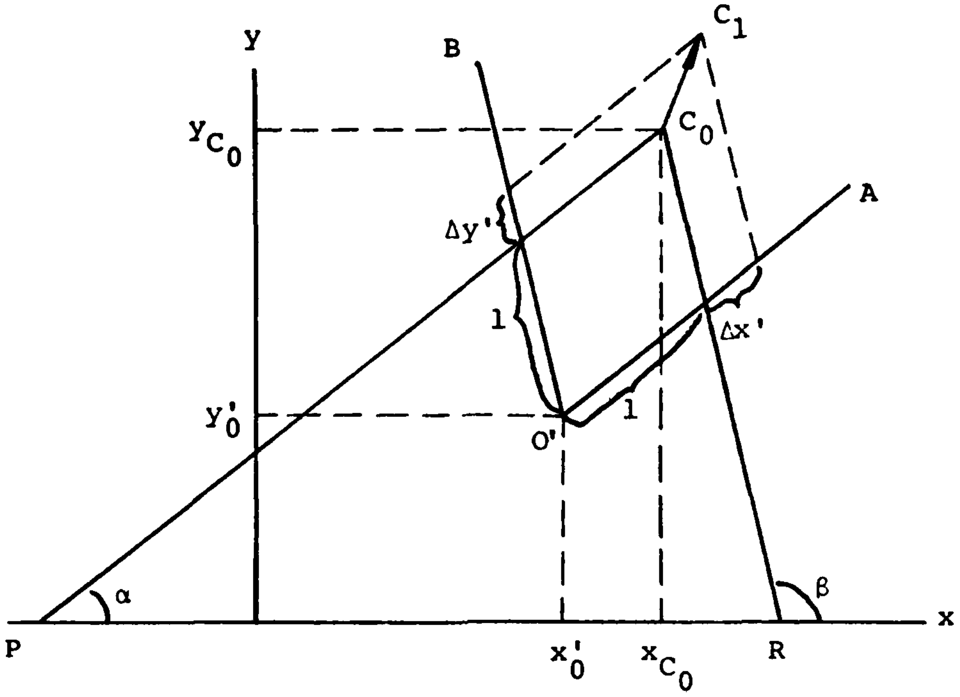


Figure 1

Now, at time t_0, the average intensity along line segments O'A and O'B is

$$I_{O'A}(t_0) = I_{O'B}(t_0) = 1/(1+m)$$

If $\Delta x'$ and $\Delta y'$ are the components of the translation in the directions of the lines O'A and O'B, then

$$I_{O'A}(t_1) = (1+\Delta x')/(1+m)$$
$$I_{O'B}(t_1) = (1+\Delta y')/(1+m)$$

assuming that m is chosen large enough so that $\max(\Delta x',\Delta y') < m$. Finally, $\Delta x'$ and $\Delta y'$ can be computed from

$$I_{O'A} = I_{O'A}(t_1) - I_{O'A}(t_0) = \Delta x'/(1+m)$$
$$I_{O'B} = I_{O'B}(t_1) - I_{O'B}(t_0) = \Delta y'/(1+m)$$

Once $\Delta x'$ and $\Delta y'$ are computed, the components of the velocity in the original image coordinate system can be recovered easily:

$$\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} = \begin{bmatrix}\cos\alpha & \cos\beta\\ \sin\alpha & \sin\beta\end{bmatrix}\begin{bmatrix}\Delta x'\\ \Delta y'\end{bmatrix}$$

The practical success of this technique depends on our ability to compute several corner parameters accurately. These parameters are

1. corner location at t_0,
2. corner shape (angles α and β), and
3. corner contrast (assumed here to be 1)

The computation of these parameters is discussed in Section 4.1. Next, we extend the previous simple analysis to include rotation as well as translation. We will treat this case as a translation from C_0 to C_1 followed by a rotation about C_1 through a clockwise angle γ (see Figure 2). Since translation and rotation are specified by a total of three parameters, we could extend the above analysis using only a third line segment parallel to either O'A or O'B. Instead, we consider two pairs of parallel line segments, and compute the displacements in the directions O'A and O'B rather than directly computing the angle γ.

Let $\Delta x'_t$, $\Delta y'_t$ be the translational components of the motion in the O'A and O'B directions, and $\Delta x'_r$ and $\Delta y'_r$ the corresponding rotational components. Then

$$\Delta x'_t + \Delta x'_r = (1+m)\ \Delta I_{O'A} \qquad (2.1)$$

$$\Delta y'_t - \Delta y'_r = (1+m)\ \Delta I_{O'B} \qquad (2.2)$$

From Figure 3, we see that

$$\frac{\Delta x'_r}{1+\Delta y'_t} = \frac{\Delta x'_r - \Delta x''_r}{\delta} \qquad (2.3)$$

where δ is the distance between the parallel line segments O'A and CD. Similarly

$$\frac{\Delta y'_r}{1+\Delta x'_t} = \frac{\Delta y'_r - \Delta y''_r}{\delta} \qquad (2.4)$$

Also

$$\Delta x'_r - \Delta x''_r = (1+m)[I_{O'A}(t_1)-I_{CD}(t_1)] \qquad (2.5)$$

$$\Delta y'_r - \Delta y''_r = (1+m)[I_{O'B}(t_1)-I_{EF}(t_1)] \qquad (2.6)$$

Substituting (2.5) and (2.6) into (2.3) and (2.4) and simplifying, we obtain

$$\Delta x'_r - C_1 \Delta y'_t = C_1 \qquad (2.7)$$

$$C_2 \Delta x'_t - \Delta y'_r = -C_2 \qquad (2.8)$$

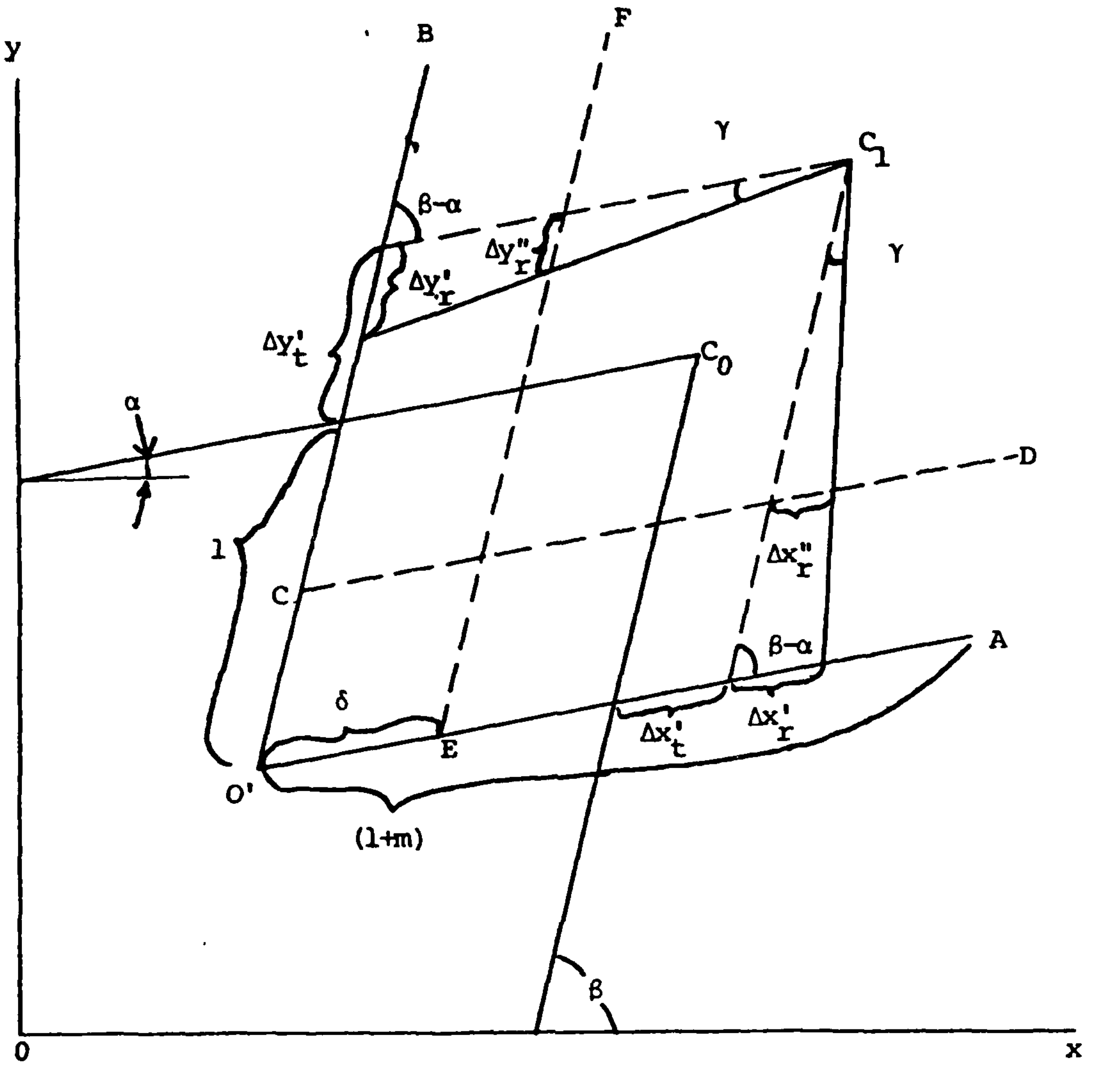


Figure 2

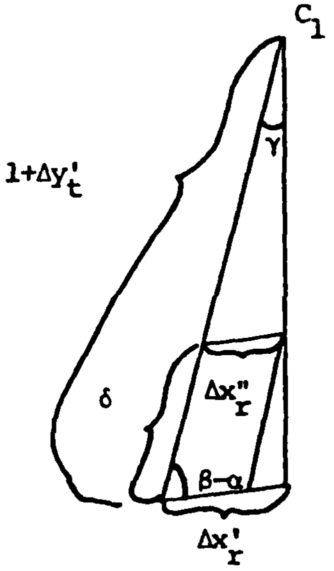


Figure 3

where

$$C_1 = \frac{(1+m)}{\delta}\,[I_{O'A}(t_1) - I_{CD}(t_1)]$$

$$C_2 = \frac{(1+m)}{\delta}\,[I_{EF}(t_1) - I_{O'B}(t_1)]$$

Solving for $\Delta x'_t$ and $\Delta y'_t$ we obtain

$$\Delta x'_t = \frac{(1+m)(\Delta I_{O'A} - C_1\Delta I_{O'B}) - C_1(1+C_2)}{1+C_1C_2} \qquad (2.9)$$

$$\Delta y'_t = \frac{(1+m)(C_2\Delta I_{O'A} - C_1\Delta I_{O'B}) - C_2(C_1-1)}{1+C_1C_2} \qquad (2.10)$$

Substituting (2.7) and (2.8) into (2.9) and (2.10), we can also compute $\Delta x'_r$ and $\Delta y'_r$, which gives us a complete description of the motion of the corner.

2.2 A least-squares approach

In this section we show how simple least-squares algorithms can be used to compute corner motion. We can rewrite (1.1) to obtain

$$V_n = -I_t/|\nabla I| \qquad (2.11)$$

where $|\nabla I|$ is the magnitude of the spatial intensity gradient at that point ($|\nabla I| = \sqrt{I_x^2+I_y^2}$) and V_n is the projection of the velocity vector $\underline{V}$ onto the intensity gradient at that point. We will first consider the case when the velocity is only a translation, and then consider translation with rotation.

If we assume that the velocities are constant in a small neighborhood of the corner along the contour, then we can relate the problem of determining the velocity $\underline{V}$ at the corner to that of determining a $\underline{V}$ to minimize

$$E = \sqrt{\{\sum_i (\underline{V}\cdot\bar{n}_i - V_{ni})^2\}} \tag{2.12}$$

where $\bar{n}_i(n_{i1}, n_{i2})$ is the unit normal vector of the i^{th} contour point and V_{ni} is the projection of $\underline{V}$ onto the intensity gradient direction at the i^{th} contour point, called the normal projection for short. By minimizing the error E^2, we obtain

$$\begin{aligned} au + cv &= d \\ cu + bv &= e \end{aligned} \tag{2.13}$$

where

$$a = \sum_{i=1}^{k} n_{i1}^2$$

$$b = \sum_{i=1}^{k} n_{i2}^2$$

$$c = \sum_{i=1}^{k} n_{i1}\cdot n_{i2}$$

$$d = \sum_{i=1}^{k} n_{i1}\cdot V_{ni}$$

$$e = \sum_{i=1}^{k} n_{i2}\cdot V_{ni}$$

From equation (2.12), we have the velocity estimation of a turning point

$$\begin{aligned} u &= \frac{bd-ce}{ab-c^2}, \\ v &= \frac{ae-cd}{ab-c^2}. \end{aligned} \tag{2.14}$$

Notice that the solutions for u and v are only meaningful when $ab-c^2$, which is related to the variance of normal directions, is high. For a straight line segment, e.g., there is no solution because the denominators of eqs. (2.13) are zero,

$$\begin{aligned} ab-c^2 &= \sum_{i=1}^{k} n_{i1}^2 \cdot \sum_{i=1}^{k} n_{i2}^2 - \left(\sum_{i=1}^{k} n_{i1} \cdot n_{i2}\right)^2 \\ &= kn_1^2 \cdot kn_2^2 - (kn_1 \cdot n_2)^2 \\ &= 0 \end{aligned}$$

i.e., from a small element of a straight line, the only information that one can obtain is the motion component normal to that line, and motion along this line element cannot be detected. Corners, however, are just those points where the variance of normal directions is locally maximal.

Now, consider the case in which the motion of the contour can be decomposed into a translation with velocity $\underline{V_0}$ (V_{0x}, V_{0y}) at the corner (x_0, y_0) and a rotation around (x_0, y_0) with the angular velocity $\underline{w}$, as shown in Figure 4. Note that since we are only interested in the motion of the corner, we do not explicitly solve for w, although it would be easy to do so.

We obtain

$$\underline{w} \times \underline{d}_i = \underline{V}_i - \underline{V}_0 \qquad (2.15)$$

By rewriting this equation, we have

$$\underline{V}_i = \begin{pmatrix} V_{0x} - wd_{iy} \\ V_{0y} + wd_{ix} \end{pmatrix} \qquad (2.16)$$

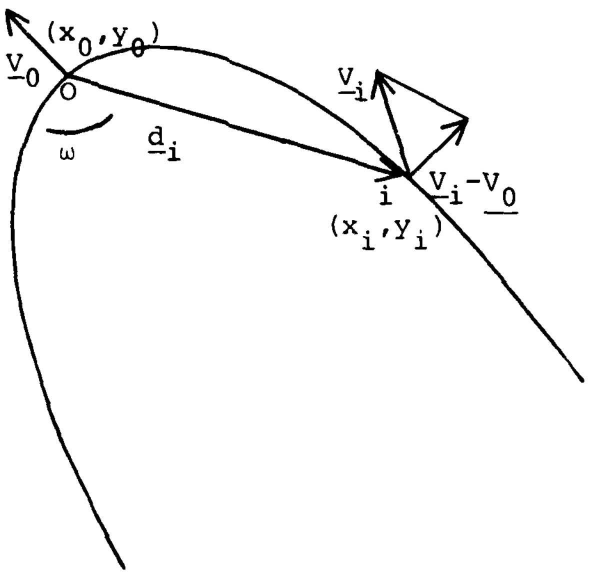


Fig. 4. Two-dimensional rigid motion.

where d_{iy}, d_{ix} are the components of the displacement vector $\underline{d}_i$ from point (x_0, y_0) to point (x_i, y_i) on the contour. The normal component of $\underline{V}_i$ is related to $\underline{d}_i$, w and $\underline{V}_0$ by

$$V_{ni} = \underline{V}_i \cdot \bar{n}_i$$

$$= (V_{0x} - wd_{iy})n_{ix} + (V_{0y} + wd_{ix})n_{iy} \qquad (2.17)$$

By considering three points on the contour, V_{0x} and V_{0y} can be simply obtained by solving linear equations. In general, more than three image points are taken and V_{0x}, V_{0y} are computed by minimizing the following square error;

$$E^2 = \sum_i (V_{0x}n_{ix} + V_{0y} \cdot n_{iy} + w(d_{ix}n_{iy} - d_{iy}n_{ix}) - V_{ni})^2$$

We obtain the following least squares solution of V_{0x}, V_{0y}:

$$V_{0x} = \frac{\begin{vmatrix} C_1 & S_{11} & a_1 \\ C_2 & S_{02} & a_2 \\ C_3 & S_{01} & a_3 \end{vmatrix}}{\delta}$$

$$(2.18)$$

$$V_{0y} = \frac{\begin{vmatrix} S_{20} & C_1 & a_1 \\ S_{11} & C_2 & a_2 \\ S_{10} & C_3 & a_3 \end{vmatrix}}{\delta}$$

where

$$S_{rpqk} = \sum_i n_{ix}^r \; n_{iy}^p \; d_{ix}^q \; d_{iy}^k$$

$$\delta_{rp} = \sum_i n_{ix}^r \; n_{iy}^p$$

and

$$a_1 = S_{1110} - S_{2001},$$

$$a_2 = S_{0210} - S_{1101},$$

$$a_3 = S_{0110} - S_{1001},$$

$$c_1 = \sum_i V_{xi} \cdot n_{ix},$$

$$c_2 = \sum_i V_{ni} \cdot n_{iy},$$

$$c_3 = \sum_i V_{ni},$$

and

$$S = \begin{vmatrix} S_{20} & S_{11} & a_1 \\ S_{11} & S_{02} & a_2 \\ S_{10} & S_{01} & a_3 \end{vmatrix}$$

3. Propagation of velocity vectors along image contours

3.1 The local constraint and the propagation formula

Suppose the velocity vectors $\underline{V_0},\underline{V_k}$ at the ends of a contour A_0A_k are known (see Figure 4). Consider a small line segment dS along the contour A_0A_1. Assuming that the motion is a rigid motion V_{0S} of V_0, the motion of A_0, parallel to A_0A_1 must equal the parallel component V_{1S} of the velocity $\underline{V_1}$ at A_1:

$$V_{0S} = V_{1S} \tag{3.1a}$$

or

$$\underline{V_0} \cdot \overline{dS} = \underline{V_1} \cdot \overline{dS}. \tag{3.1b}$$

where $\underline{V_0}$ and $\underline{V_1}$ are the velocity vectors at the two ends of the line dS, and $\overline{dS}$ is the unit vector along dS, the vector joining A_0 to A_1. Rewriting this local constraint (eq. 3.1b) into component form, we obtain

$$\begin{aligned} \underline{V_0} \cdot \overline{dS} &= (V_{1n}\overline{n} + V_{1t}\overline{t}) \cdot \overline{dS} \\ &= V_{1n}\overline{n} \cdot \overline{dS} + V_{1t}\overline{t} \cdot \overline{dS} \end{aligned} \tag{3.2}$$

where V_{1n} and V_{1t} are the normal component and the tangential component of the velocity vector $\underline{V_1}$ respectively, and $\overline{n}$ and $\overline{t}$ are the unit vectors in the normal and tangent directions of the contour at A_1. From Figure 5, we see that

$$\begin{gathered} \overline{n}\cdot\overline{dS} = \cos\alpha \text{ and } \overline{t}.\overline{dS} = \cos(\pi/2-\alpha) = \sin\alpha \text{ so that} \\ V_{0S} = V_{1t}\sin\alpha + V_{1n}\cos\alpha \end{gathered} \tag{3.3}$$

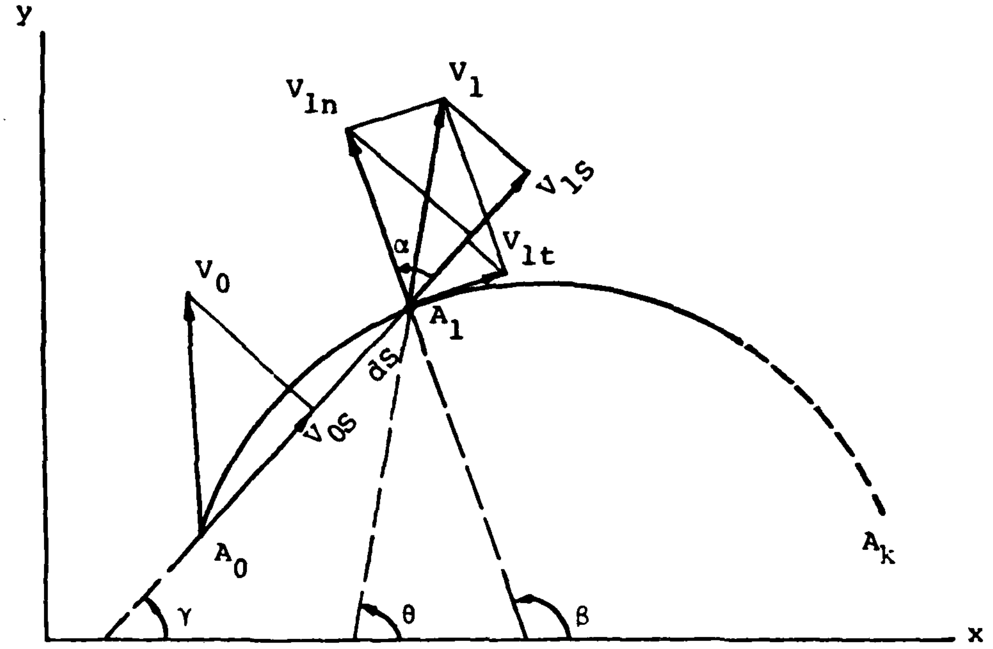


Figure 5. The geometry of the propagation along a contour in an image.

Thus, we have that the tangential component is

$$V_{1t} = (V_{0S} - V_{1n} \cdot \cos\alpha)/\sin\alpha \qquad (3.4)$$

where α is the angle between the unit vector $\overline{dS}$ and the normal vector $\bar{n}$ at the point A_1. We also have $\gamma=\beta-\alpha$, where β is the angle between the x-axis and the normal vector $\bar{n}$, and γ is the angle between the x-axis and the line segment dS.

We can propagate the velocity along a contour using eq. (3.4), because the first projection V_{0S} is known after the previous propagation and the normal component V_{1n} can be computed by, e.g., the methods discussed in [3] or [4]. Once V_{1n} is computed, $\underline{V_1}=V_1e^{j\theta}$ can be obtained because

$$V_1 = \sqrt{V_{1n}^2+V_{1t}^2} \qquad (3.5)$$

$$\theta = \beta - \arctan V_{1t}/V_{1n}$$

3.2 Error analysis and a correction technique

From eq. (3.4) the new estimate of the tangent component V_{1t} is based on the previous projection V_{0S} and on the normal component V_{1n} at the current propagation point. Differentiating this equation we obtain

$$dV_{1t} = \frac{\partial V_{1t}}{\partial V_{0S}}\, dV_{0S} + \frac{\partial V_{1t}}{\partial V_{1n}}\, dV_{1n} + \frac{\partial V_{1t}}{\partial \alpha}\, d\alpha$$

$$= \frac{1}{\sin\alpha}\, dV_{0S} - \cot\alpha\, dV_{1n} + \frac{(V_{1n} - V_{0S}\cos\alpha)}{\sin^2\alpha}\, d\alpha \qquad (3.6)$$

Note that the error in V_{1t} depends on the error in the previous projection (dV_{0S}), the error in the normal component V_{1n} at the current propagation point (dV_{1n}), and the error in the measurement of the angle α ($d\alpha$).

The result of these various errors is that when the propagation reaches A_k, the velocity vector $\underline{V_k'}$ attributed to A_k by the propagation procedure will differ from the velocity vector originally computed at A_k. Therefore, at the point A_k we compute the error between the propagation velocity estimate $\underline{V_k'}$ and the original velocity vector $\underline{V_k}$ and compute the error

$$\underline{\Delta V_k} = \underline{V_k} - \underline{V_k'}$$

If this error is less than some tolerance, then this propagation procedure is stopped at point A_k; otherwise a correction procedure is applied. If we consider the error ΔV_k as having been accumulated in the previous n steps, then the average velocity error in one step is

$$\underline{V_e} = \underline{\Delta V_k}/k$$

so we have $m \cdot \underline{V_e}$ as the velocity error at the m^{th} step and we propagate this velocity error step by step backward to correct the estimated velocity vector at each point along the same contour.

4. Experimental results

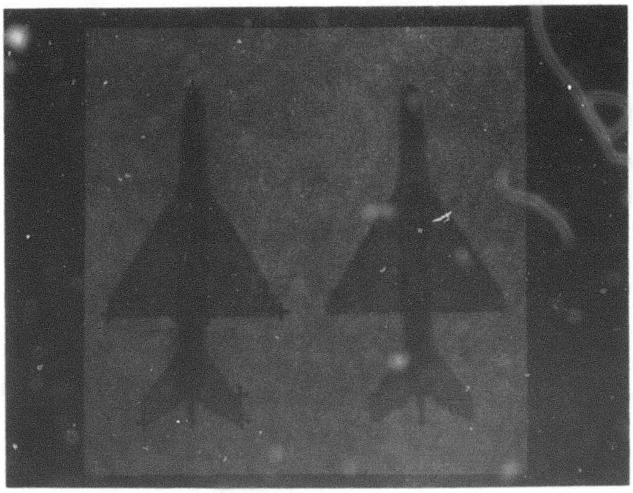
We applied the corner motion estimation and velocity propagation algorithms to two sets of motion pictures. Section 4.1 describes the corner motion estimation results, and Section 4.2 describes the propagation results.

4.1 Corner motion estimation

The three corner motion models described in Section 2 were applied to two image sequences containing two frames each (Figures 6-7).

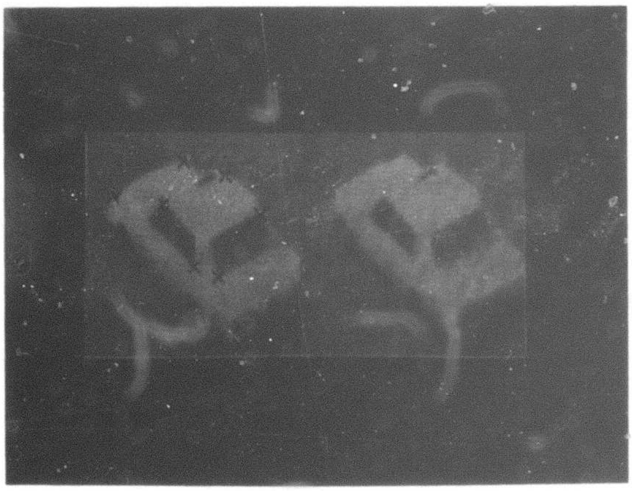
We first describe the application of the structural model presented in Section 2.1. Corners are "provisionally" detected using the corner detection algorithm described in Kitchen and Rosenfeld [6]. Next, a small window around each corner is analyzed to obtain a more accurate description of the corner. Based on the assumption that the corner locally contrasts with its surround, a local thresholding procedure (Milgram [7]) is used to segment the window. The corner is then relocated to a maximum curvature boundary point in the thresholded window. The slopes of the line segments meeting at the corner are computed using a one-dimensional (slope) Hough transform procedure (only slope need be computed since the lines are constrained to pass through the corner point.) The corners detected by this procedure are marked with dark crosses in Figures 6a and 7a.

To overcome the effects of various sources of error on the motion estimation, several quadruples of line segments are used to compute estimates of $\Delta x'_t$, $\Delta y'_t$, $\Delta x'_r$ and $\Delta y'_r$, with the final motion estimate taken as the average.



(a) (b)

Figure 6. Airplane sequence.



(a) (b)

Figure 7. Traffic sequence.

The results for the airplane in Figure 6 are displayed in Table 1. The estimated motion vectors were obtained by the authors' examination of digital enlargements of the images. No useful results were obtained for the moving car in Figure 7. There are several reasons for this:

1. The grey level corners in the car are much more rounded than the airplane's, and the motion estimates are sensitive to the corner location; and
2. The spatial resolution of Figure 7 is not high enough to allow us to place a sufficiently large window around a corner for segmentation which does not contain some other image feature.

The least-squares corner motion estimates presented in Section 2.2 require that we first compute the normal component of motion along the contour in the neighborhood of the corner. The magnitude of the normal component and the components of the unit normal vector on the x and y axes are

$$V_n = -I_t/\sqrt{I_x^2+I_y^2} \tag{4.1}$$

$$n_1 = I_x/\sqrt{I_x^2+I_y^2} \tag{4.2}$$

$$n_2 = I_y/\sqrt{I_x^2+I_y^2}\ d \tag{4.3}$$

The derivatives (I_x,I_y,I_t) in (4.1)-(4.3) are approximated as follows:

$$I_x \doteq \frac{1}{4} \sum_{\ell=0}^{1} \sum_{m=0}^{1} \{I_{k+\ell,i+m,j+1} - I_{k+\ell,i+m,j}\}$$

$$I_y \doteq \frac{1}{4} \sum_{\ell=0}^{1} \sum_{n=0}^{1} \{I_{k+\ell,i+1,j+n} - I_{k+\ell,i,j+n}\}$$

$$I_t \doteq \frac{1}{4} \sum_{m=0}^{1} \sum_{n=0}^{1} \{I_{k+1,i+m,j+n} - I_{k,i+m,j+n}\}$$

The unit of length is the grid spacing interval in each image frame and the unit of time is the image frame sampling period.

Tables 2 and 3 contain the results of applying both least-squares corner estimators to the images in Figures 6 and 7, respectively.

					measured		estimated	
	x	y	alpha	beta	u	v	u	v
1	39	93	170.1	213.5	-8.4	-1.5	-1.2	-5.9
2	166	46	264.5	345.6	-0.2	-0.8	-0.2	-1.0
3	166	146	16.4	96.6	-2.7	-1.8	-3.2	-1.5
4	167	87	89.5	185.4	-2.3	-1.6	-1.0	-0.9
5	168	104	175.1	276.6	-1.7	-1.2	-2.1	-0.6
6	212	123	39.2	195.3	-0.7	1.6	-2.7	0.6
7	227	68	9.6	292.6	-1.0	0.7	-1.0	1.3
8	227	124	70.2	353.8	-2.8	1.0	-2.9	1.0

Table 1. Motion vectors for corners in Figure 6.

		estimated		computed by (2.14)		computed by (2.18)	
x	y	u	v	u	v	u	v
39	93	-1.2	-5.9	3.1	-2.7	2.7	-0.5
166	46	-0.2	-1.0	-0.1	-0.8	0.1	-0.7
166	146	-3.2	-1.5	-1.2	-0.8	-0.2	-1.7
167	87	-1.0	-0.9	-0.9	-0.8	-1.0	-0.9
168	104	-2.1	-0.6	-1.8	-0.7	-2.4	-0.6
212	123	-2.7	-0.6	-1.7	1.0	-3.8	0.5
227	68	-1.0	1.3	-0.9	1.2	-1.0	0.6
227	124	-2.9	1.0	-2.4	0.1	-3.5	3.7

Table 2. Motion vectors for turning points of Figure 6 (airplane)
(1) Under local translation assumption.
(2) Under local planar motion assumption.

		estimated		computed by eq. (2.14)		computed by eq. (2.18)	
x	y	u	v	u	v	u	v
9	29	-1.5	-0.2	-1.4	-0.03	-1.4	-0.1
19	9	-1.5	-0.3	-1.3	-0.1	-1.4	-0.1
61	46	-1.8	-2.2	-1.1	-1.7	-1.8	-2.4
56	46	-1.6	-2.2	-1.5	-1.5	-0.8	-2.7
23	52	-1.6	-1.0	-1.0	-1.3	-1.0	-1.3
13	41	-1.2	-0.6	-1.2	-0.5	-1.2	-0.6
21	24	-1.8	-1.5	-1.4	-0.5	-1.6	-1.8
39	33	-1.8	-1.5	-1.7	-1.1	-1.4	-2.5
41	40	-1.3	-1.5	-2.1	-1.2	-0.4	3.0

Table 3. Motion vectors for turning points of Figure 7 (traffic).
(1) Under local translation assumption.
(2) Under local planar motion assumption.

4.2 Velocity propagation

We applied the propagation technique to the two image sequences displayed in Figures 6 and 7.

The propagation technique was implemented as follows:

1) Velocity vectors are first determined at a set of "corner" points in the first frame by the least-square corner motion estimator which assumes that the corner motion is a 2-D translation.

2) The velocity vector at the corner is propagated along the contours that meet at the corner until a second corner point is encountered. The contours are followed by a very simple maximum gradient technique. A velocity vector is not computed at every pixel on the contour, but only at every k^{th} pixel, to reduce the error in α.

3) When the terminating corner point is reached, the propagation is stopped and the error velocity vector is computed. If this error is greater than a preset tolerance, then the error velocity vector is back-propagated along the same contour.

The results of the propagation are displayed in Figures 8-9.

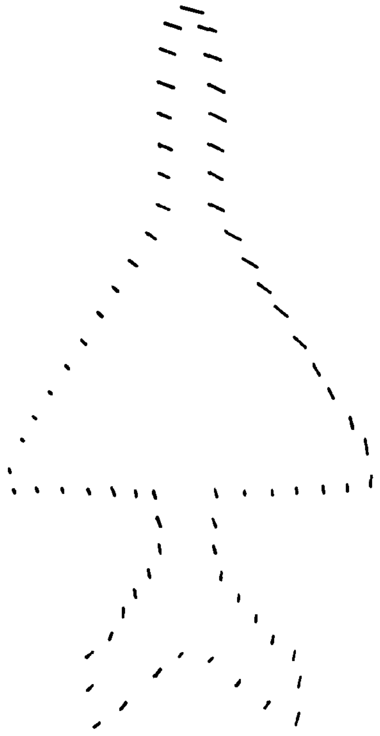


Figure 8. Velocity field using the propagation technique along the contours of the moving airplane shown in Figure 6.

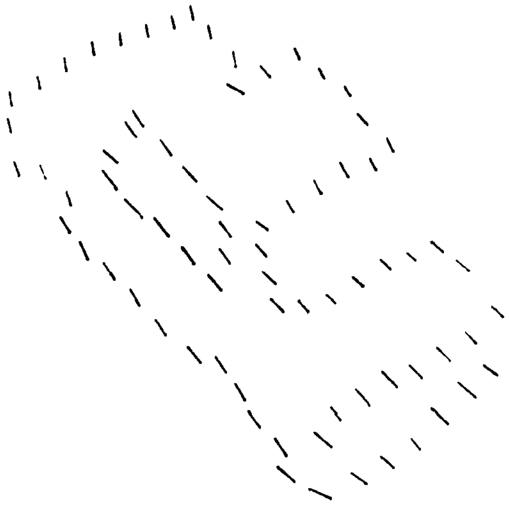


Figure 9. Velocity field using the propagation technique along the contours of the moving car shown in Figure 7.

5. Summary

We have presented a contour-based approach to motion estimation based on first estimating motion at image corners and then propagating these motion estimates along image contours. One potential advantage of such an approach over others such as [3-4] is that motion information is not integrated across the boundaries of moving objects, but only along such boundaries. Since very often the only reliable source of motion information is at object boundaries (when, for example, object interiors are homogeneous) it is important that motion estimation techniques yield accurate motion estimates at boundaries.

The examples presented in Section 4 both consisted of a single object moving across a homogeneous background. The propagation technique presented in Section 3 would need to be modified to be applicable to more complex image sequences containing multiple moving objects and occlusions so that motion information is not propagated from one object to another.

References

1. Ullman, S., "Analysis of visual motion by biological and computer systems," Computer, 14, 8, 57-69, 1981.

2. Thompson, W. B. and S. T. Barnard, "Lower-level estimation and interpretation of visual motion," Computer, 14, 8, 20-28, 1981.

3. Glazer, F., "Computing optical flow," Proc. 7th Int. Joint Conf. Artificial Intelligence, Vancouver, B.C., 644-647, August 1981.

4. Horn, B. K. P. and B. G. Schunk, "Determining optical flow," A.I., 17, 1981, 185-204.

5. Nagel, H. H., "Displacement vectors derived from second order intensity variations in image sequences, Feb. 1982.

6. Kitchen, L. and A. Rosenfeld, "Gray level corner detection," University of Maryland Computer Science TR-887, April 1980.

7. Milgram, D., "Region extraction using convergent evidence," Comp. Graphics and Image Processing, 11, 1-12, 1979.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 2. GOVT ACCESSION NO. AD-A124 812	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CONTOUR-BASED MOTION ESTIMATION	5. TYPE OF REPORT & PERIOD COVERED Technical
	6. PERFORMING ORG. REPORT NUMBER TR-1179
7. AUTHOR(s) Larry S. Davis Zhongquan Wu Hanfang Sun	8. CONTRACT OR GRANT NUMBER(s) DAAG-53-76C-0138
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Vision Laboratory Computer Science Center University of Maryland College Park, MD 20742	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Night Vision Lab. Ft. Belvoir, VA 22060	12. REPORT DATE June 1982
	13. NUMBER OF PAGES 28
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Image processing
Pattern recognition
Time-varying imagery
Motion estimation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper introduces a contour-based approach to motion estimation. It is based on first computing motion at image corners, and then propagating the corner motion estimates along the principal contours in the image based on a local $2\frac{1}{2}$D motion assumption. The results of several experiments are presented.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)